Question

Explain what the following does, and find the time complexity:

```
void findWhatIDo(BST *r1, BST *r2)
{
if ( (r2 == NULL) || (r1 == NULL) ) return;

int h1 = GetHeight(r1->root);
int h2 = GetHeight(r2->root);

BST *bn1 = h1 > h2 ? r1 : r2 ;
BST *bn2 = (bn1 == r1) ? r2 : r1;
BST_Node *tmp = bn2->root;

while(!bn2->IsEmpty())
{
tmp = bn2->ExtractRoot();
bn1->Insert(tmp);
}
}
```

Reminder about syntax of ternary operator:
(condition)?(statement to execute if true):statement to execute if false);

Hint: Inserting a node in a binary search tree with n nodes has the same run time complexity as a binary search over n nodes

Microsoft Interview Question for Software Engineer / Developers. (n.d.).

Solution:

This merges two binary search trees.
Time complexity = 0(n1log(n2)), where n1 is number of nodes in the tree with lesser height, and n2 is number of nodes in tree with greater height.